

## IN THE CLAIMS:

1. A method for accessing data in a first memory based database environment (MBE), said method comprising the steps of:
  - receiving a processing request;
  - 5 retrieving an update from a data chain associated with the first MBE;
  - constructing a message for an MBE using the retrieved update; and
  - 10 sending the message to a second MBE.
2. The method according to claim 1, wherein said data chain is a chronological data chain.
3. The method according to claim 1, further comprising the step of:
  - 15 writing the retrieved update to an update queue file, if the second MBE is not available.
4. The method according to claim 1, wherein the data chain is an internal chain that includes a latest modified version of a data record of the first MBE.
5. The method according to claim 1, wherein the data chain is used to process
  - 20 modified records without a need to queue or move said modified records in the first MBE.
6. The method according to claim 1, wherein the second MBE is selected from a group consisting of: a mirrored system, a database, or a disk.
7. The method according to claim 1, wherein said first MBE is a shared-nothing environment.
8. A method for accessing data in a first memory based database environment (MBE), said method comprising the steps of:
  - 30 receiving a processing request from a disk process emulator ("DPE") process;
  - locking a data segment of a DPE; and
  - retrieving an update from a data chain.

9. The method according to claim 8, wherein said data chain is a chronological data chain.
10. The method according to claim 8, wherein the steps are accomplished by an MBE  
5 synchronization process ("MSP").
11. The method according to claim 8, further comprising the step of:  
constructing a message for a second MBE.
- 10 12. The method according to claim 11, wherein said second MBE environment is  
selected from a group consisting of: a mirrored system, a database, or a disk.
13. The method according to claim 11, further comprising the step of:  
unlocking the data segment.
- 15 14. The method according to claim 13, further comprising the step of:  
sending the message to the second MBE, if it is available.
15. The method according to claim 14, further comprising the step of:  
20 waiting for a completion message from the second MBE.
16. The method according to claim 13, further comprising the step of:  
writing updates to an update queue file, if the second MBE is not available.
- 25 17. The method according to claim 16, wherein there are more than one update queue  
files, further comprising the step of:  
using a queue control file, on restart if the MSP fails, to determine which update  
queue file needs to be processed, and to determine a state of said update  
queue files.
- 30 18. The method according to claim 8, wherein the data chain is an internal chain that  
includes a latest modified version of a data record of the first MBE.

19. The method according to claim 8, wherein the data chain is used to process modified records without a need to queue or move said modified records in the first MBE.

20. The method according to claim 8, wherein the first MBE is a shared-nothing  
5 environment.

21. A method for accessing data in a memory based database environment (MBE), said method comprising the steps of:

10 receiving by an MBE file support process ("MFS"), startup information and configuration information from a manager process ("MGR");  
receiving by said MFS, processing requests from a disk process emulator ("DPE") process;  
locking a data segment in response to the processing requests; and  
retrieving an update from a data chain.

15

22. The method according to claim 21, wherein said data chain is a chronological data chain.

20 23. The method according to claim 21, further comprising the steps of:  
constructing an internal update buffer of pending changes; and  
unlocking the data segment.

24. The method according to claim 23, further comprising the steps of:  
performing a database modification to a specified file;  
25 awaiting the next request from a DPE; and  
using the DPE control segment for temporary storage of bundles of updates.

25. An apparatus for accessing data in a first memory based database environment ("MBE"), said apparatus comprising:  
30 a disk process emulator, said disk process emulator maintaining a chronological data chain; and

an MBE synchronization process, wherein said MBE synchronization process uses said chronological data chain to send database modifications to a second MBE environment on behalf of said disk process emulator.

26. The apparatus of claim 25 wherein said chronological data chain is an internal chain that includes the latest modified version of a data record.

27. A computer-readable medium comprising instructions to cause a computer in a  
5 memory based database environment (MBE) to:  
receive processing requests from a disk process emulator ("DPE") process;  
lock a data segment of the requesting DPE; and  
retrieve a pending update from a data chain.

10 28. The computer-readable medium according to claim 27, wherein said data chain is a chronological data chain.

29. The computer-readable medium according to claim 27, wherein the data chain is an internal chain that associates the latest modified version of a data record.

15

30. The computer-readable medium according to claim 27, wherein the data chain is used to process modified records without the need to queue or move said modified records.